In the Claims:

Please amend the claims as follows. The following listing of claims replaces all previous

listings.

1 – 11 (Cancelled).

12. (Currently Amended) A computer program product having a non-transitory computer

readable medium embodying program code executable by a computing device, ~~with computer~~

~~program logic recorded thereon~~, said computer program product comprising:

code for initializing of a rich internet application (RIA), wherein initializing comprises

instantiating, at a computing device accessing the computer readable medium, at least one visible

element in said RIA and deferring instantiation of unseen ones of a plurality of stacked elements

in said RIA;

code for generating, at the computing device, a descriptor tree having a plurality of

descriptor nodes, wherein each of said plurality of descriptor nodes describes an ~~a plurality of~~

instantiated and visible interface element[[s]] of said RIA;

code for creating one or more stacked descriptor nodes in said descriptor tree describing

said unseen ones of said plurality of stacked interface elements responsive to a user navigating to

said unseen ones, wherein said unseen ones are not instantiated at said starting; and

code for rendering a graphical representation of said plurality of visible interface

elements and said unseen ones at a display connected to the computing device, the graphical

representation rendered by using the descriptor tree to identify ~~using~~ corresponding ones of:

said plurality of descriptor nodes; and

said one or more stacked descriptor nodes.

13. (Previously Presented) The computer program product of claim 12 further comprising:

code for converting said plurality of descriptor nodes into a plurality of detail objects;

code for converting said one or more stacked descriptor nodes into one or more stacked detail objects, wherein said plurality of instantiated and visible interface elements and said unseen ones are rendered directly using said plurality of detail objects and said one or more stacked detail objects.

14. (Original) The computer program product of claim 12 wherein each one of said plurality of descriptor nodes and said one or more stacked descriptor nodes contains a software method for generating each its child nodes.

15. (Previously Presented) The computer program product of claim 12 further comprising:

code for downloading bytecode representing said RIA to a computer of said user responsive to said initializing of said RIA, wherein said code for generating and said code for creating use said bytecode.

16. (Previously Presented) The computer program product of claim 13 further comprising:

code for storing as a plurality of stored nodes each of:

said plurality of descriptor nodes;

said one or more stacked descriptor nodes; said plurality of detail objects; and

said one or more stacked detail objects; and

code for re-rendering each of said plurality of instantiated and visible interface elements and said one or more stacked interface elements from said plurality of stored nodes.

17. (Original) The computer program product of claim 12 wherein said one or more stacked descriptor nodes created has a navigational relationship with a particular one of said one or more stacked interface elements to which said user navigates.

18. (Original) The computer program product of claim 17 wherein said navigational relationship comprises one or more of:

a direct link;

an ordinal relationship;

a statistical relationship; and a positional relationship.

19. (Previously Presented) The computer program product of claim 12 further comprising:

code for creating select ones of said one or more stacked descriptor nodes in said descriptor tree responsive to starting said RIA.

20 – 27. (Cancelled).

28. (Currently Amended) A computer program product having a <u>non-transitory</u> computer readable medium <u>embodying program code executable by a computing device,</u> ~~with computer program code recorded thereon~~, said computer program product comprising:

program code for accessing<u>, at a computing device,</u> executable code of a rich internet application, the executable code comprising code for instantiating a plurality of objects, each object for rendering a corresponding interface element of the rich internet application;

program code for creating<u>, at the computing device and in response to accessing,</u> a descriptor tree comprising a plurality of descriptor nodes, each node identifying an object or container of the application<u>, the descriptor tree created based on the accessed code</u>;

program code for identifying, at the computing device, a subset of the plurality of objects in the executable code, the subset comprising fewer than all of the plurality of objects, wherein identifying the subset comprises (i) using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view and (ii) excluding the object from the subset of objects;

program code for instantiating, at the computing device, the objects in the subset in response to identifying the subset of objects;

program code for rendering, at a display connected to the computing device, a graphical representation showing an initial view of the application, the graphical representation rendered using the instantiated subset of objects;

program code for instantiating, at the computing device, at least one other object of the plurality of objects in response to user interaction with an interface element of the initial view; and

program code for rendering, at the display connected to the computing device, a second graphical representation showing another view of the application, the second graphical representation rendered using the instantiated at least one other object.

29. (Cancelled).

30. (Previously Presented) The computer program product set forth in claim 28 , wherein using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view comprises:

determining if the node identifies itself as corresponding to a stacked navigation object.

31. (Previously Presented) The computer program product set forth in claim 28 , wherein creating a descriptor tree comprising identifying a node whose object is for rendering an interface element that is not visible in the initial view as a hidden node; and

wherein objects associated with hidden nodes are excluded from the subset.

32. (Currently Amended) A method comprising:

accessing, by a computing device, code of a rich internet application from a computer-readable medium;

in response to accessing, generating, by the computing device, a descriptor tree based on the accessed code, the descriptor tree having a plurality of descriptor nodes corresponding to respective elements of the rich internet application, at least some of the descriptor nodes identifying interface elements of the rich internet application intended to be instantiated and visible at a beginning of the rich internet application;

creating, by the computing device, at least one hidden descriptor node in the descriptor tree, the hidden node identifying an interface element not intended to be instantiated or visible at the beginning of the rich internet application;

instantiating, by the computing device, and rendering a plurality of interface elements at the beginning of the rich internet application based on identifying corresponding non-hidden nodes in the descriptor tree;

rendering a first graphical representation using the instantiated plurality of interface elements;

responsive to data identifying navigation in the rich internet application, instantiating, by the computing device and rendering a second interface element based on a corresponding hidden

descriptor node <u>and rendering a second graphical representation using the instantiated second</u> <u>interface element,</u>

wherein instantiation of the second interface element <u>and rendering the second graphical</u> <u>representation</u> [[is]] <u>are</u> deferred until the navigation occurs.

33. (Previously Presented) The method set forth in claim 32, wherein each of the descriptor nodes contain a software method for generating each of its child nodes, if any.

34. (Previously Presented) The method set forth in claim 32, further comprising

prior to accessing the executable code, downloading the code.

35. (New) The computer program product set forth in claim 12, wherein the descriptor tree comprises a hierarchical data structure comprising an application descriptor node, the application descriptor node including a child constructor method for creating a child node of the application descriptor node, the child constructor method comprising code logic containing a pointer to code of the rich internet application.

36. (New) The computer program product set forth in claim 28, wherein the descriptor tree comprises a hierarchical data structure comprising an application descriptor node, the application descriptor node including a child constructor method for creating a child node of the application descriptor node, the child constructor method comprising code logic containing a pointer to code of the rich internet application.

37. (New) The method set forth in claim 32, wherein the descriptor tree comprises a hierarchical data structure comprising an application descriptor node, the application descriptor node including a child constructor method for creating a child node of the application descriptor node, the child constructor method comprising code logic containing a pointer to code of the rich internet application.

38. (New) A system comprising:

a display device;

a memory; and

a processor,

wherein the memory embodies program code that configures the processor to:

access code of a software application,

based on the accessed code, generate a descriptive hierarchical structure of the application, the descriptive hierarchical structure comprising a plurality of descriptor nodes, the descriptive hierarchical structure comprising at least one first descriptor node that corresponds to an object used in rendering in an initial view of the application and at least one second descriptor node that corresponds an object used in rendering a second view of the application other than the initial view of the application,

use the descriptive hierarchical structure to identify the at least one first descriptor node and instantiate an object corresponding to the at least one first descriptor node, and

render an initial view of the application by using the instantiated object to generate a graphical display for output using the display device.

39. (New) The system set forth in claim 38, wherein the program code further configures the processor to:

receive input navigating to the second view,

instantiate a second object corresponding to the at least one second node.

40. (New) The system set forth in claim 38, wherein the descriptive hierarchical structure includes nodes corresponding to objects of a plurality of different views of the application.